Presenter: Chase Petersen

Members: Brandon French, Kevin Krohn, Boston Knighton-Johnson

public async Task<List<ImageModel>> GetAllImagesAsync()

{

//Creating a table of images by using a list

var imageTableResults = new List<ImageModel>();

//Creating a table that queries through the object we give it. In this case it's an image.

TableQuery<ImageModel> tableQuery = new TableQuery<ImageModel>();

//Has to equal null because it cannot be a generic value

TableContinuationToken continuationToken = null;

//Checks the condition at the end of loop, basically saying just do this at least once

do

{

//This is getting particular results of a certain query. It executes the table as tableQuary and

//where we are in the table via continuationToken

TableQuerySegment<ImageModel> tableQueryResult =

await this.cloudTable.ExecuteQuerySegmentedAsync(tableQuery, continuationToken);

//Sets to where we left off

continuationToken = tableQueryResult.ContinuationToken;

//Adds that data that we retrieved from the cloudTable from the tableQuary only where the upload was completed

imageTableResults.AddRange(tableQueryResult.Results.Where(result => result.UploadComplete));

//Checks to see if we are still getting continuation tokens which means we are still getting quary results

} while (continuationToken != null);

//Returns the results of the query

return imageTableResults;

}